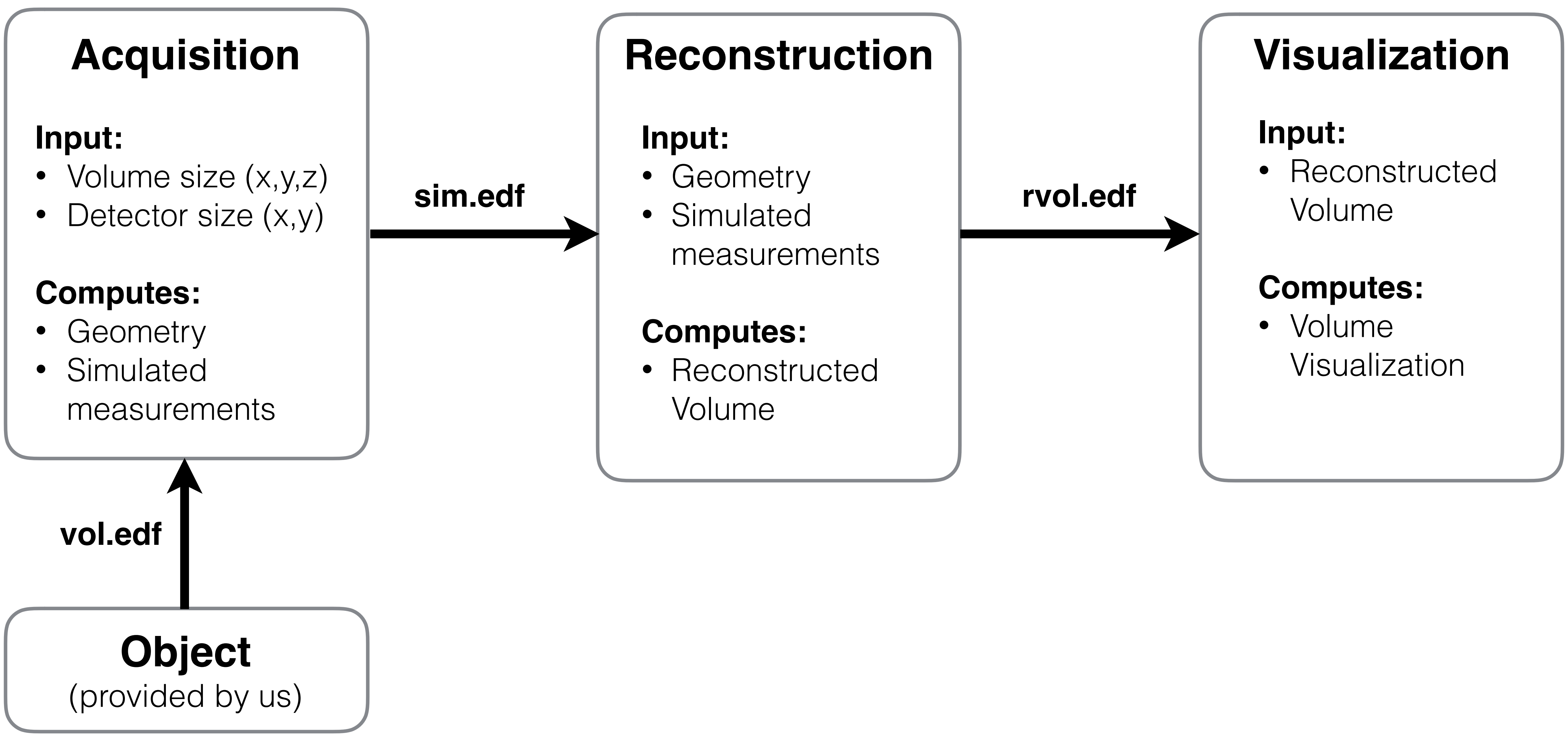
Part 2:
Image Acquisition

Tobias Lasser

Part 2: Image Acquisition

- Part 2: Goals & deadline
- Computed tomography: motivation
- Forward model
- Your task

Part 2: Flow-diagram



Part 2: Deadline

- Deadline for all 3 subprojects: **Wednesday, February 1, 2017**

- each team has been assigned a git repository Ws16Cpp/part2-team?
- your project has to build successfully using our gitlabCI runners!
- at final event on **February 7, 2017** we will:
    - execute your main program on our build server computer
        - ./yourprogram mystery.edf
    - we expect to see:
        - visualization of geometry and simulated measurements
        - control of the reconstruction parameters and progress indicator of reconstruction process
        - volume visualization of the reconstruction results

Part 2: Reconstruction

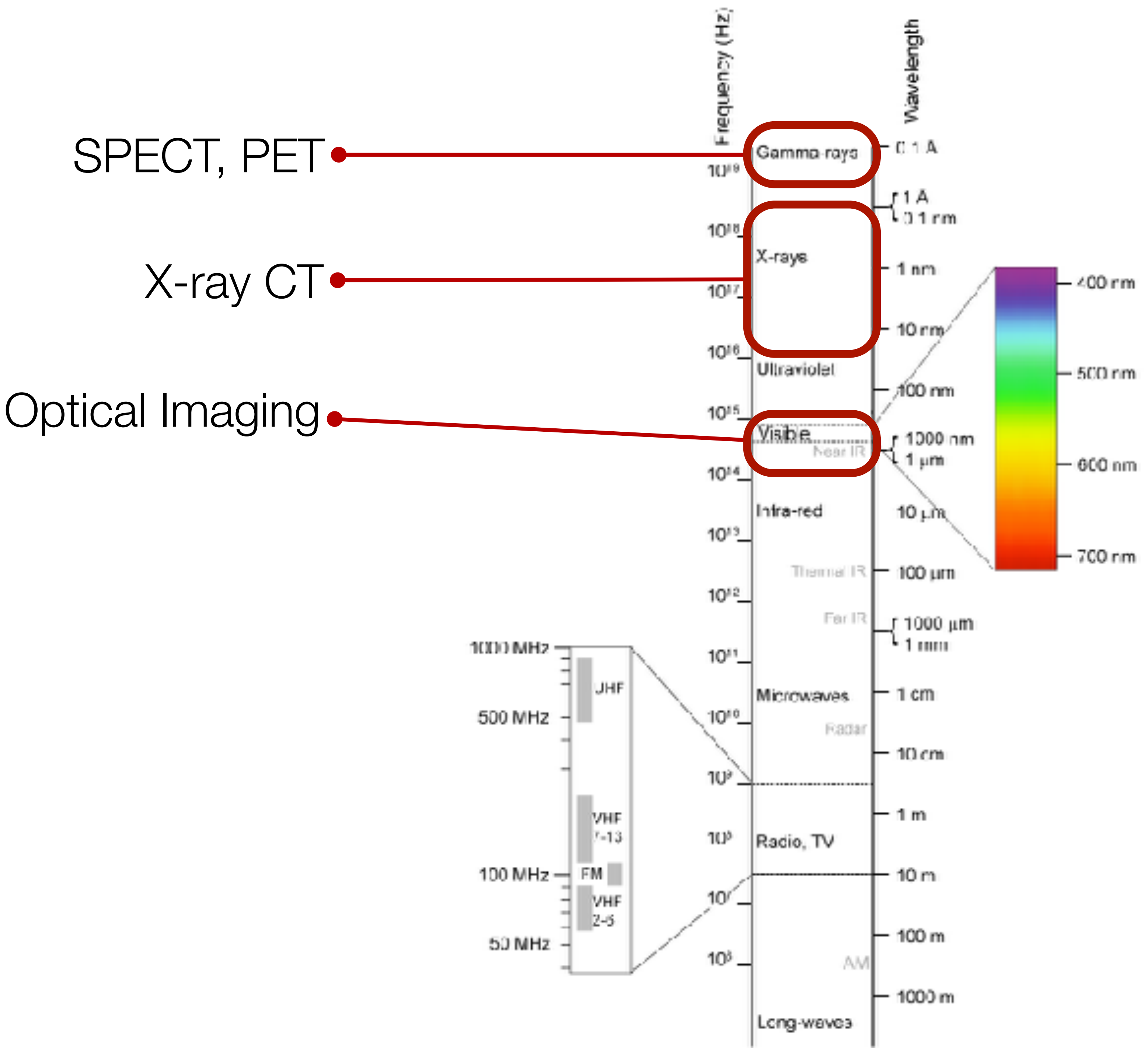
- Part 2: Goals & deadline
- Computed tomography: motivation
- Forward model
- Your task

X-ray imaging and Computed Tomography

X–ray image of the hand of Anna Bertha Röntgen, 1895.

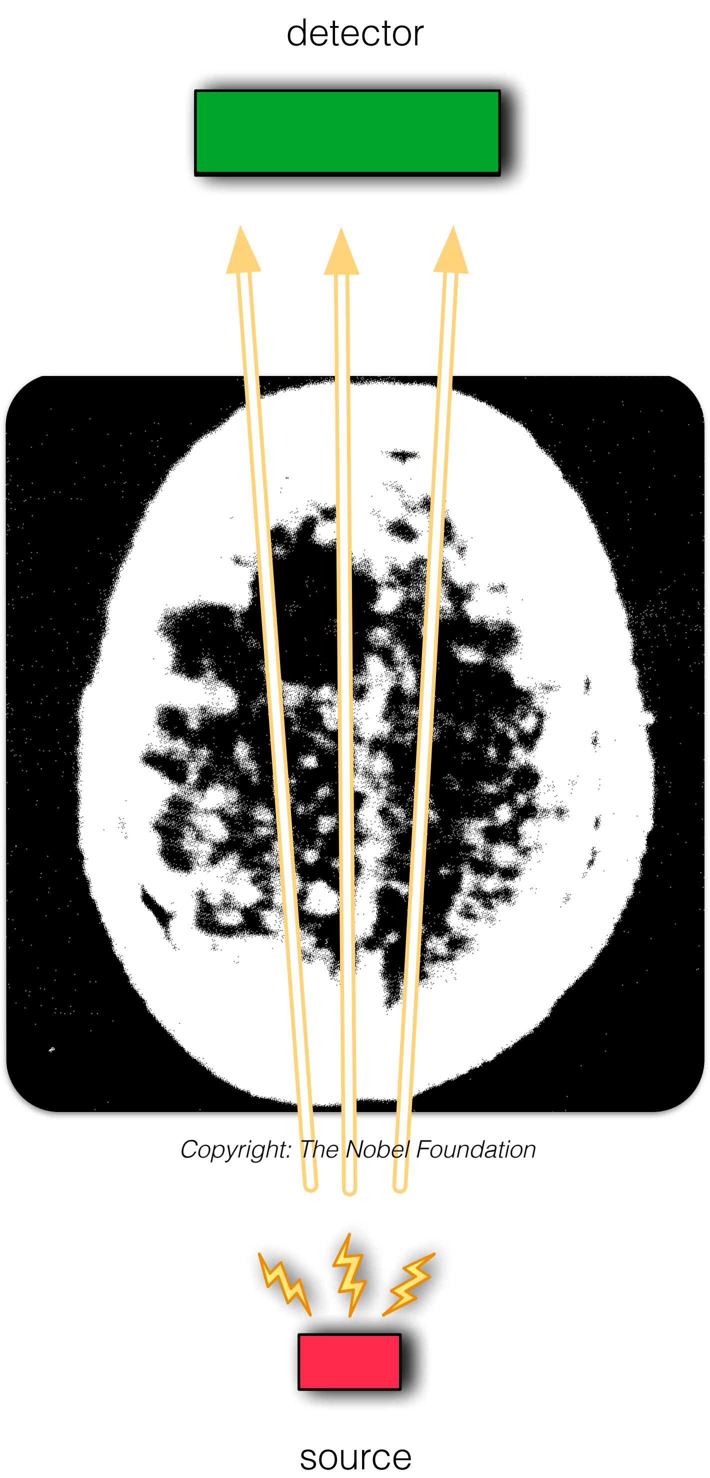
First clinical CT slice of a woman with a suspected brain lesion, 1972.

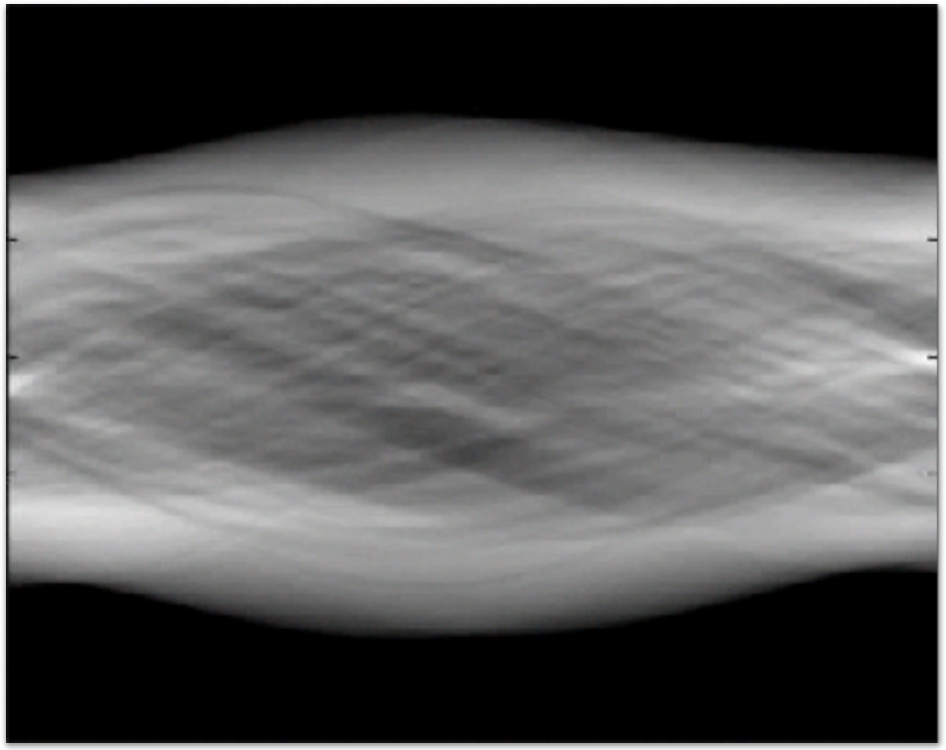
A bit of physical background



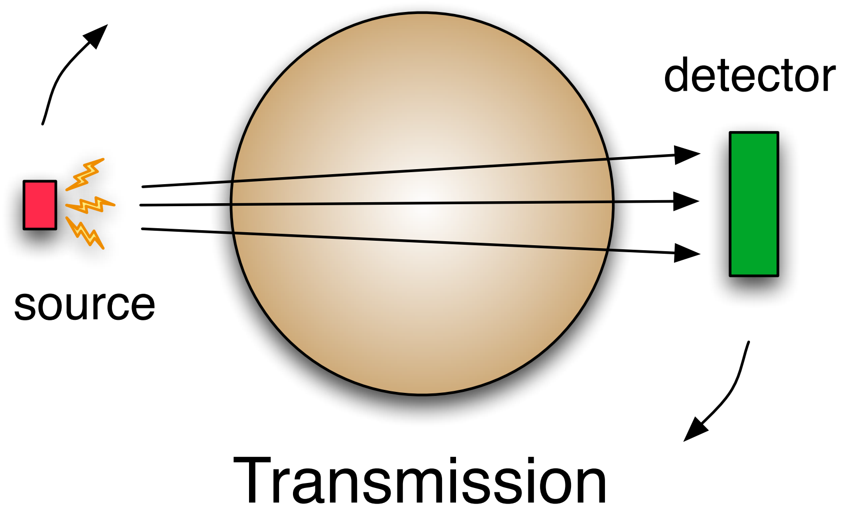
Source: wikipedia, author: user Materialscientist, license: CC-by-sa/2.5

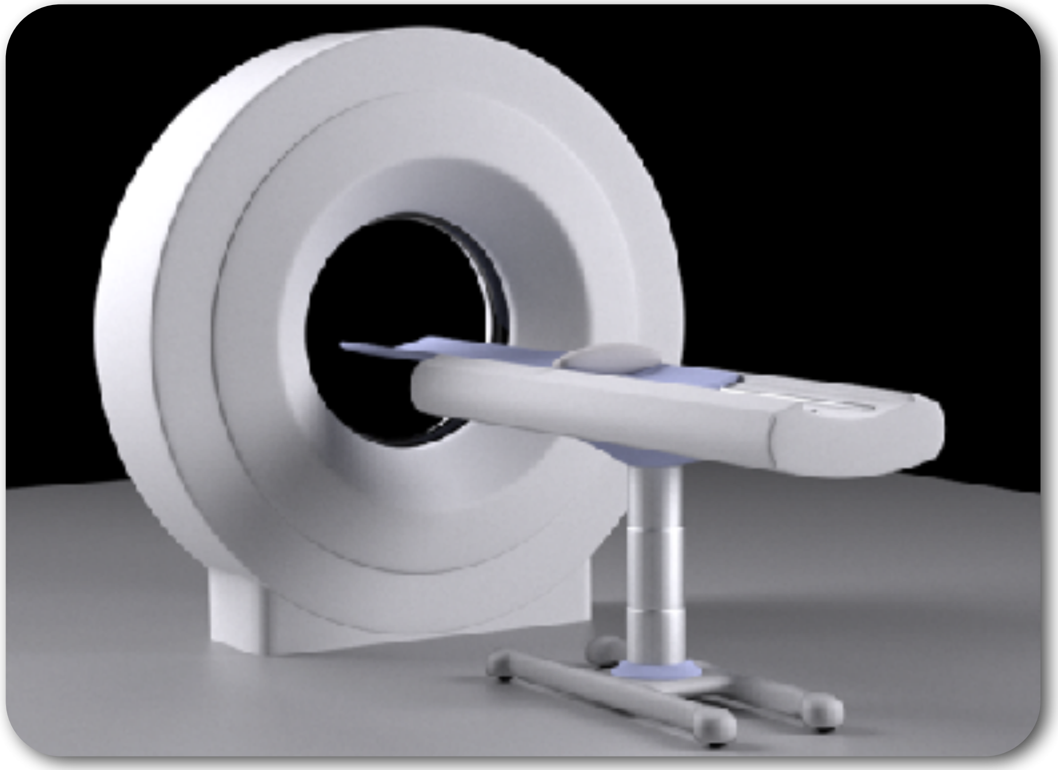
X-ray imaging and Computed Tomography





X-ray imaging for CT



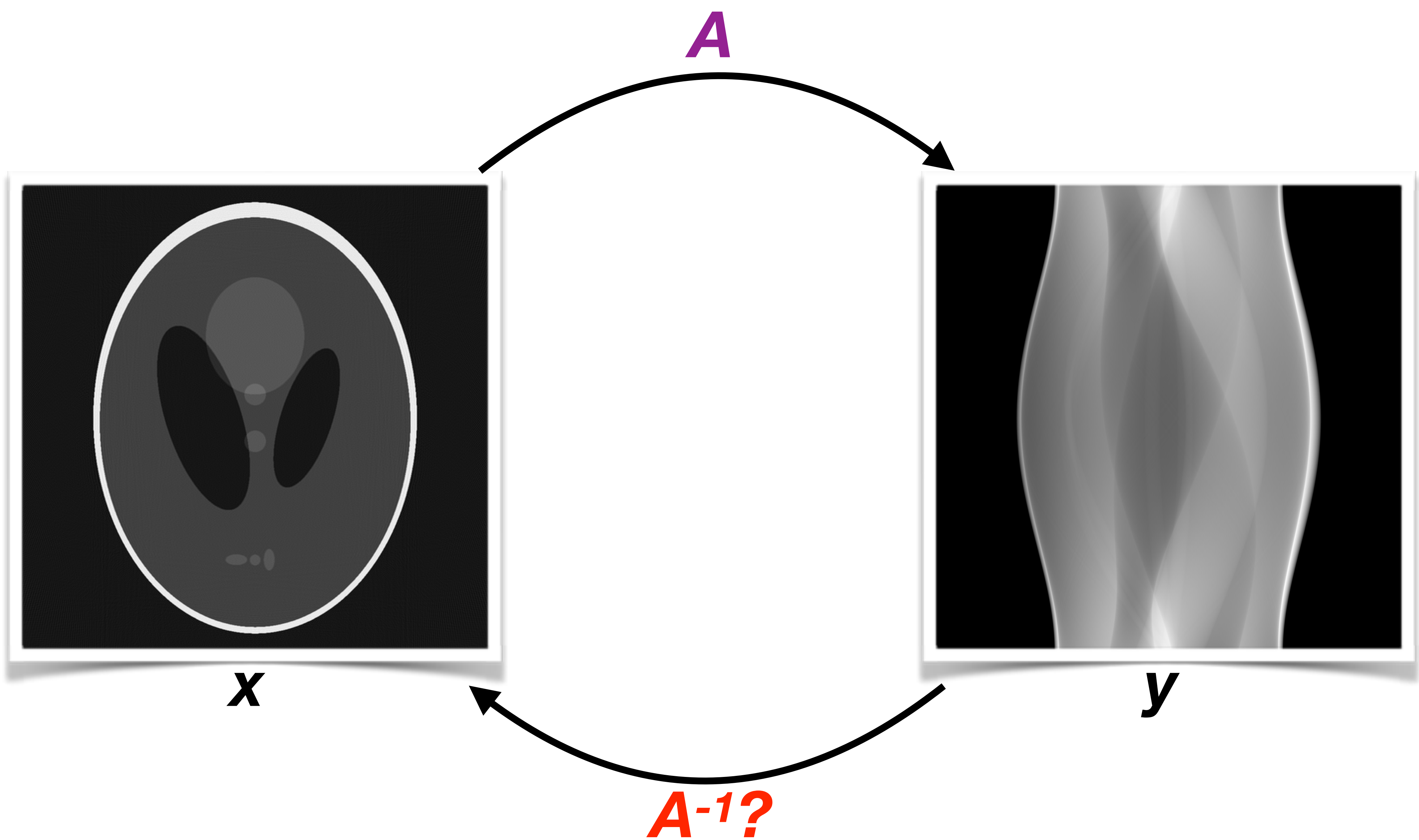


Schematic rendering of CT scanner. *Courtesy A. Duliu*

Photograph of intra-operative C-arm. *Courtesy L. Wang*

How to compute tomographic reconstructions?

Model and inversion



Part 2: Reconstruction

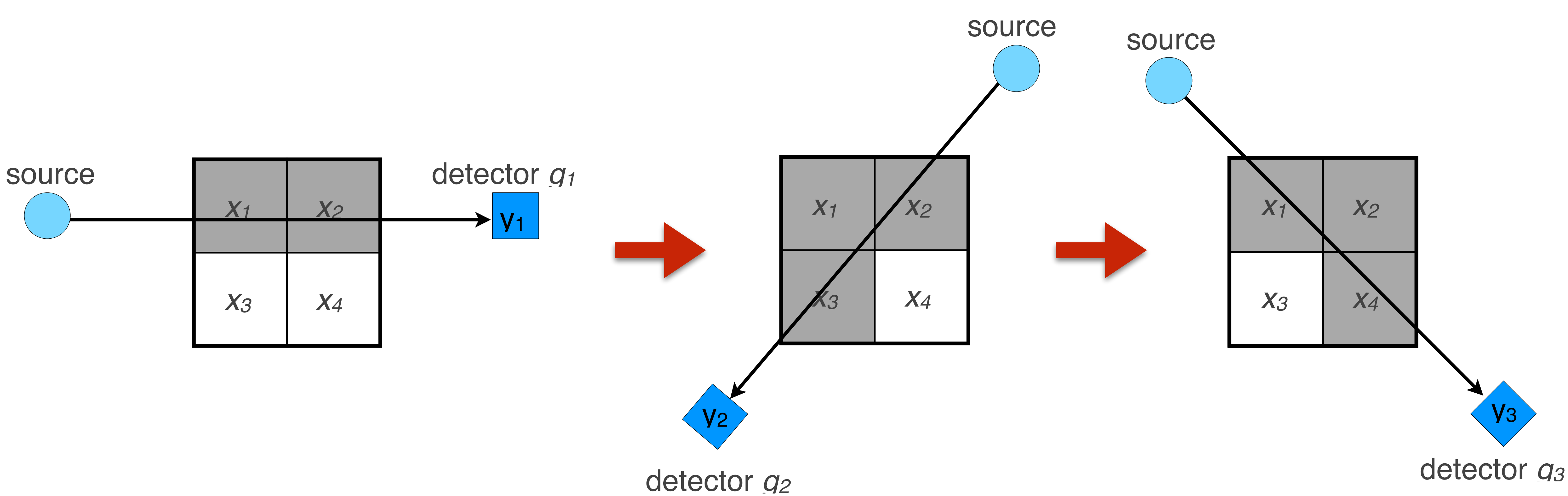
- Part 2: Goals & deadline
- Computed tomography: motivation
- Forward model
- Your task

Motivation: X-ray transform

$$y = \int_L f(x)\ dx, \qquad L \text{ line}$$

Discretization & simplified model

Volume of interest $V = \{x_1, x_2, x_3, x_4\}$. $f : V \to \mathbb{R}$ attenuation coefficient.



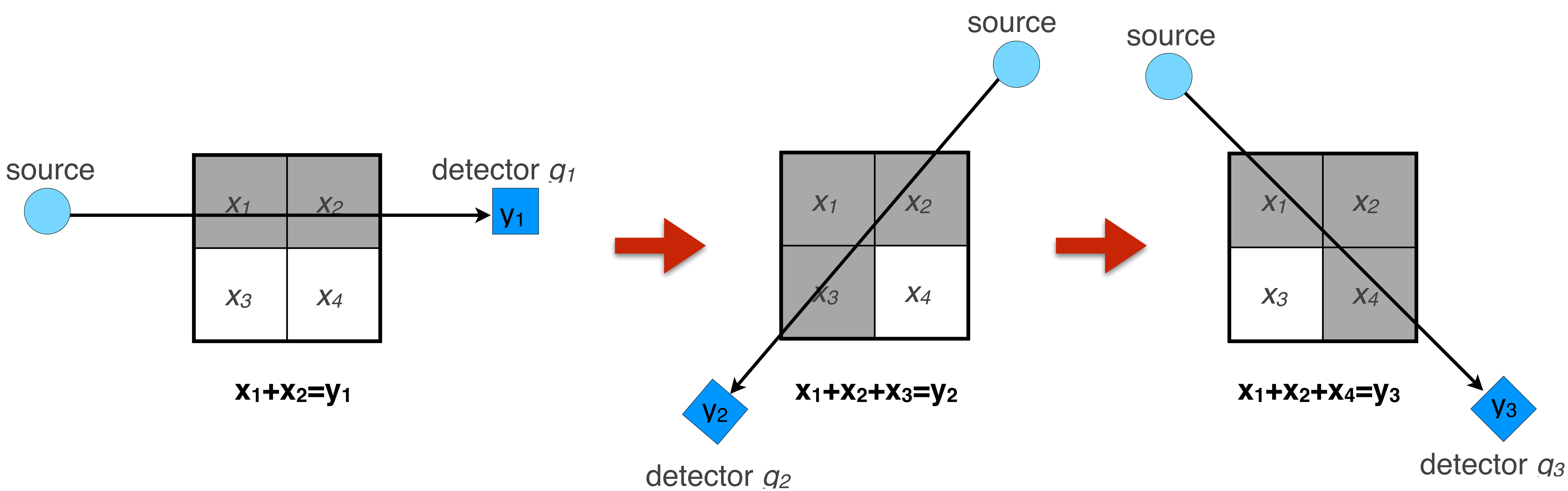
$x_1+x_2=y_1$

$x_1+x_2+x_3=y_2$

$x_1+x_2+x_4=y_3$

Discretization & simplified model

Volume of interest $V = \{x_1, x_2, x_3, x_4\}$. $f : V \to \mathbb{R}$ attenuation coefficient.



$x_1+x_2=y_1$ **$x_1+x_2+x_3=y_2$** **$x_1+x_2+x_4=y_3$**

$$\underbrace{\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}}_{\text{system matrix}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}}_{\text{volume}} = \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}}_{\text{measurements}}$$

On-the-fly system-matrix

- System matrix gets very very big (3D CT: up to several Exabyte)
- Does not fit into our memory!

- Compute multiplication with system-matrix A on the fly
    - Use raytracing
    - implement functions for computation of
        - Ax (called **forward projection**) and
        - $A^T z$ (called **back projection**)

Raytracing

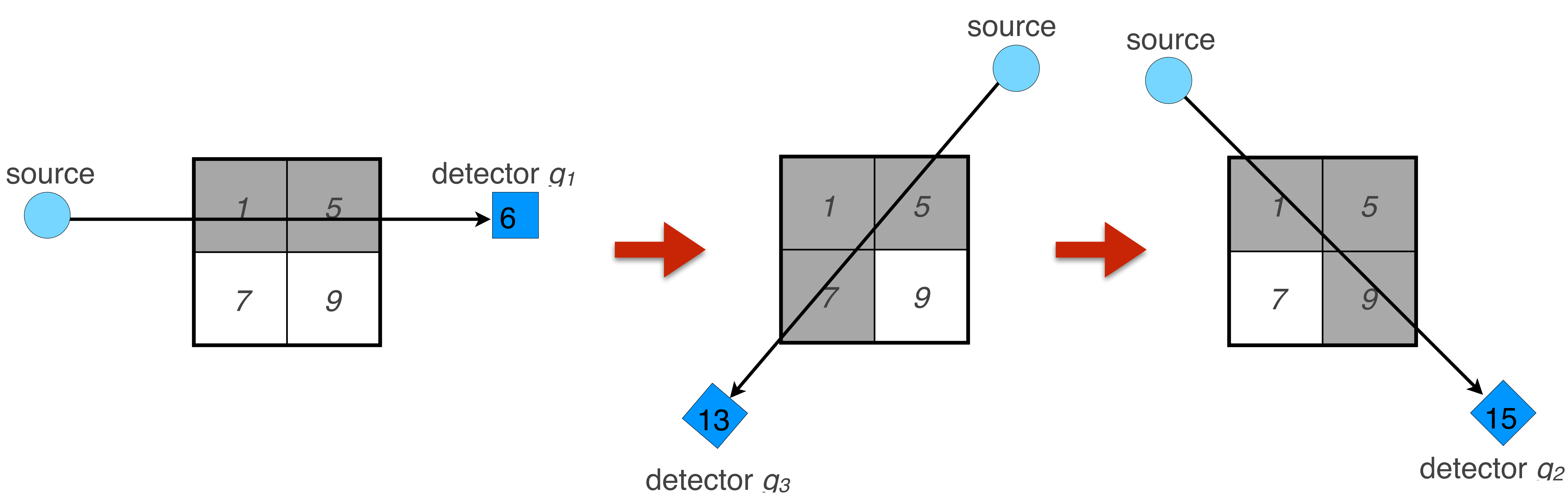
- For all rays:
  - Compute entry point, i.e. first intersection of ray and volume [1]
  - iterate over all voxels hit by the ray [2]
  - Do something with those voxels (see next slides)

[1] A. Williams, S. Barrus, R. K. Morley, and P. Shirley, “An efficient and robust ray-box intersection algorithm,” presented at the ACM SIGGRAPH 2005 Courses, New York, New York, USA, 2005, p. 9.
[2] J. Amanatides and A. Woo, “A fast voxel traversal algorithm for ray tracing,” Eurographics, 1987.

What to do with those voxels?

Forward projection: Computation of *Ax*

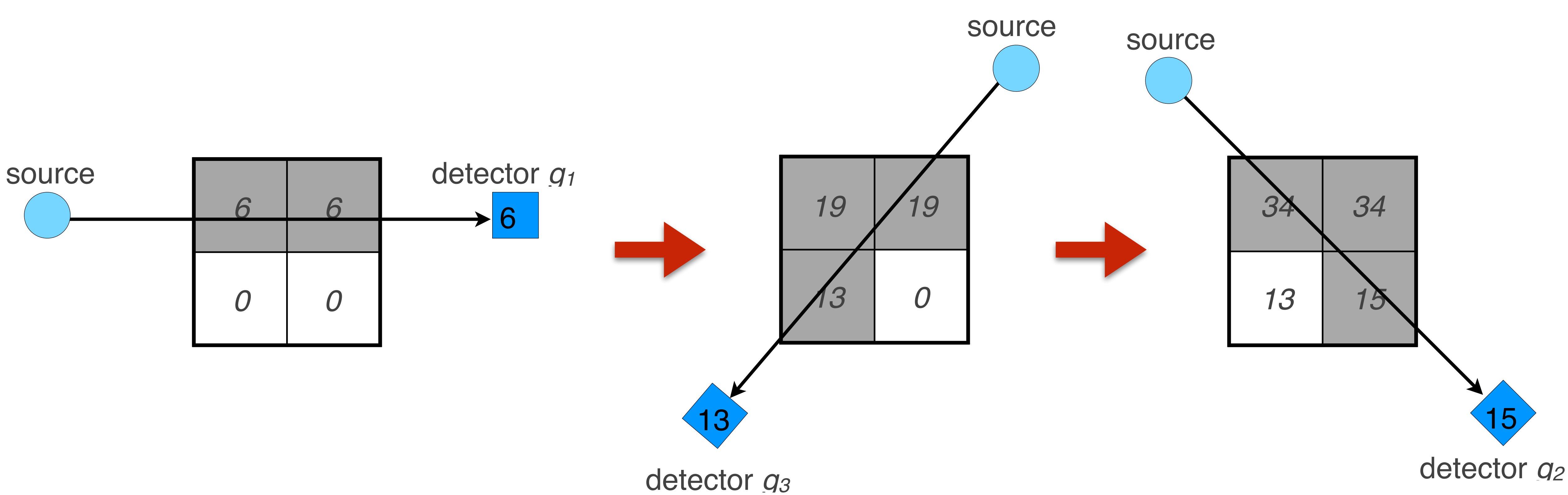
- Accumulate all voxel values along the ray



What to do with those voxels?

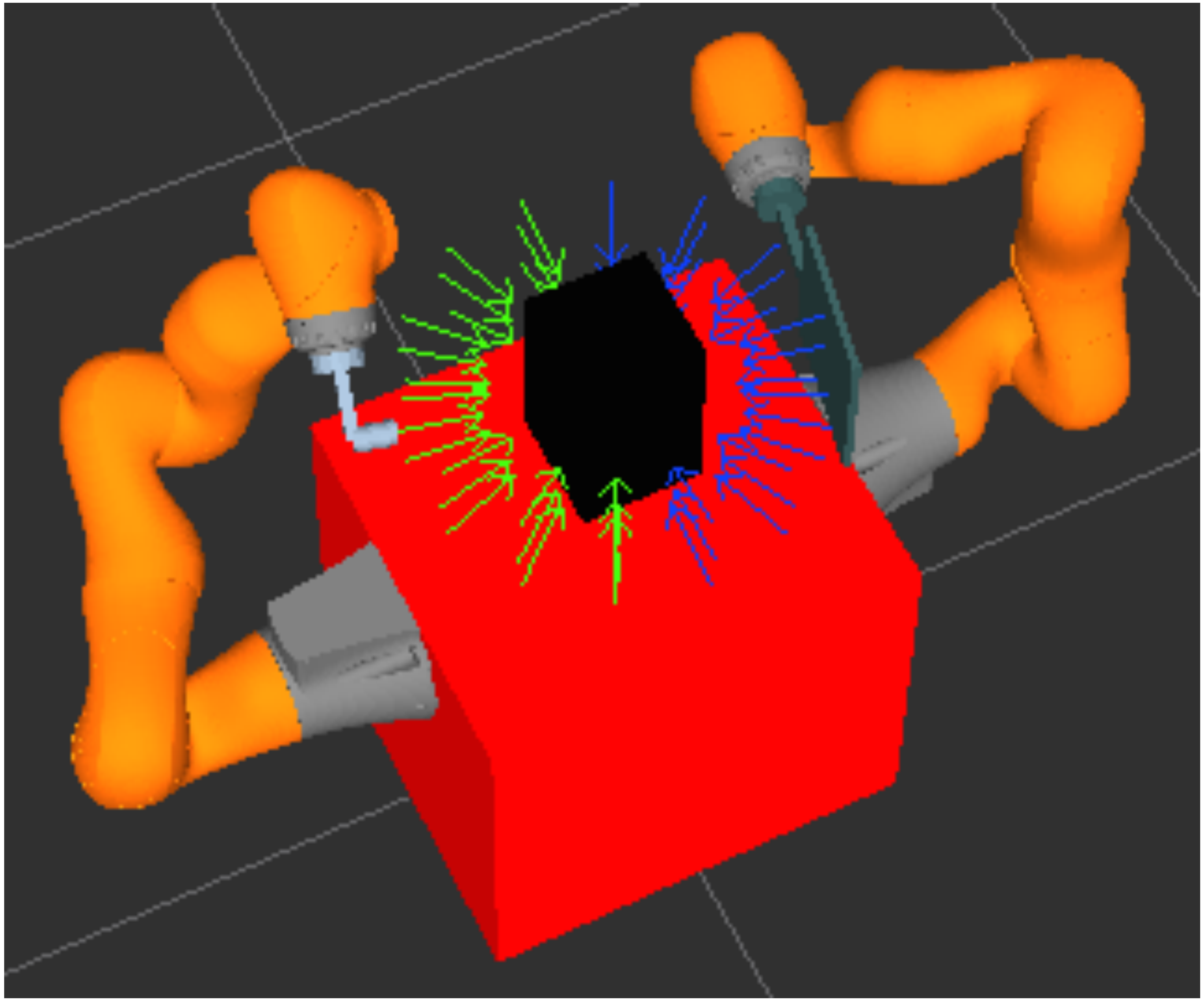
Back projection: Computation of $A^T z$

- Initial: Zero volume
- Successively add the element value of *z* to each value of a voxel hit by ray



Part 2: Reconstruction

- Part 2: Goals & deadline
- Computed tomography: motivation
- Forward model
- Your task



Your task

- implement volume handling (with .edf input/output)
- implement acquisition poses (source and detector position and orientation)
- implement ray-tracing
  - using ray/volume intersection and voxel traversal
- implement forward projection using ray-tracing
- do X-ray simulation using the forward projection
- visualize your acquisition poses and simulated X-ray images using Qt

Your task

- ask questions early and often!
- commit and push your code early and often!

- your code **has to build successfully** using all our gitlab CI runners!

- we are **not automatically testing** your code's functionality like in part 1!
    - this is your responsibility now

Demo

Questions?